

FEDERATED AUTHENTICATION SERVICE

TECHNICAL FIELD

5 The present invention relates generally to secure use of computerized networks, and more particularly to efficiently maintaining security in information systems when multiple authentication types and sources are used.

BACKGROUND ART

10

In today's digital world, information systems and their contents are among the most valuable of an organization's assets. Every year organizations spend significant amounts of money to protect their data from unauthorized access. Simultaneously, organizations have an overarching business requirement to share information with their partners, customers, suppliers, and even in some cases competitors and adversaries. This requires authentication.

The greatest value in authentication is when it forms the basis for enforcing access control rules. That is, in order for a system to determine what a subject can do the system must first ascertain who the subject is.

25

Traditional authentication systems generally presume a single authentication source and type. For example, in Kerberos the authentication source is a trusted key distribution center (KDC) and the authentication type is user IDs with passwords. [Version five of Kerberos supports initial authentication based on public keys, but a high percentage of commercial implementations of Kerberos authenticate based on a user ID and a password.] Another example is the public key infrastructure (PKI) system. Here the authentication source is a certificate authority (CA) and the authentication type is challenge/response. While both Kerberos and PKI permit multiple authentication sources, these authentication sources must be closely coupled. Often, this translates to complex trust relationships between the sources of authentication, which leads to solutions that are operationally infeasible and economically cost-prohibitive.

30

An emerging authentication system, and one which has particular importance later in this discussion is the secure remote password (SRP) protocol. In the words of SRP's advocates, inventor Tom Wu and Stanford University, "it solves the problem of authenticating clients to

servers securely, in cases where the client must memorize a small secret (like a password) and carries no other secret information, and where the server carries a verifier which allows it to authenticate the client but which, if compromised, would not allow someone to impersonate the client." But SRP, like traditional authentication systems, also presumes a single authentication source and type.

5 A practical view of inter- and intra-organization communication reveals that there can never be a single authentication type. In fact, according to a report published in February 2001 by the Giga Group, companies will be supporting multiple authentication types, such as 10 passwords, tokens, certificates and smart cards. Therefore, security architectures should include a single infrastructure for managing all of the authentication types, rather than a separate infrastructure for each. Even if there someday is a single authentication type (e.g. biometrics), there will always be multiple authentication sources, each having administrative control over a set of subjects.

It follows that information systems that seek to enforce access control must be prepared to accept authentication information from any number of sources. Indeed, key criteria for enforcing access control include the exact source and type of authentication. In a practical model the many authentication sources need to form a federation, each of whose members can ascertain the authenticity of a set of subjects.

What is needed is a technology that permits organizations to leverage authentication sources that belong to themselves, or to their customers, partners, suppliers, or any other third party. We can term such a technology a Federated Authentication Service Technology (FAST), and define its goal to be to enable organizations to quickly implement their business relationships through highly secure information systems.

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

DISCLOSURE OF INVENTION

Accordingly, it is an object of the present invention to provide a Federated Authentication Service Technology (FAST), to enable organizations to quickly implement their business 5 relationships through highly secure information systems.

Another object of the invention is to provide an authentication system which permits the use of multiple authentication types and multiple authentication sources from different domains of control.

Another object of the invention is to provide an authentication system which permits an 10 organization to change its authentication mechanism without affecting its users or servers.

Another object of the invention is to provide an authentication system which provides stringent security requirements while leveraging an organization's existing security infrastructure to quickly implement business relationships.

And another object of the invention is to provide an authentication system which permits users and service providers to choose where to authenticate.

Briefly, one preferred embodiment of the present invention is a system for authenticating a subject residing in a subject domain on a network to a server application residing in a server domain on the network when an authentication mechanism residing in an authentication domain on the network affects the service provided by the server application. The system includes a client to communicate with other components and to authenticate the subject by providing a client name assertion on behalf of the subject. The client also resides in the subject domain. The system further includes a protocol proxy to communicate between the client and the authentication mechanism and authenticate the client based on the client credentials, and to create from the client credentials an authentication name assertion allowing the client to access 25 the server application.

An advantage of the present invention is that it permits multiple authentication types and sources by effectively abstracting these. Existing authentication technologies allow multiple authentication types (e.g., user id/password, biometrics, digital certificates, etc.). However, these existing technologies do not abstract multiple authentication sources (e.g., employer, financial 30 institution, healthcare provider, etc.).

Another advantage of the invention is that it can authenticate with any mechanism server

may choose. The authenticating mechanism of a server is completely independent of that of a client.

Another advantage of the invention is that it permits flexible credential expiration, requested by the client, the server application, or the authentication mechanism.

5 Another advantage of the invention is that it supports authentication from different domains, permitting a single, global sign-on.

Another advantage of the invention is that it provides a single location for managing credentials, providing easy administration and enabling the deployment of highly secure systems.

10 Another advantage of the invention is that it employs a highly secure inter-component protocol. This enables many different deployment scenarios and provides a basis for managed security services.

Another advantage of the invention is that it is authentication mechanism agnostic. It can protect an organization's investment in existing authentication mechanisms, yet permit seamless integration of future authentication mechanisms.

15 Another advantage of the invention is that it permits a hierarchy of trust. The invention requires authenticating mechanisms to authenticate themselves. In this manner a user need only reveal sensitive credentials to authentic mechanisms. Furthermore, a server application need only trust assertions of an authentic mechanism.

20 Another advantage of the invention is that it permits production and verification of signatures using Name Assertions. This eliminates the need to use digital certificates for production and verification of signatures, which improves the speed and efficiency of signature production and verification.

25 Another advantage of the invention is that it allows name assertions to be used as a basis to renew an existing name assertion. This eliminates the need to use digital certificates every time authentication is needed, which improves on the speed and efficiency of the authentication process.

Another advantage of the invention is that it is standards compliant. It promotes interoperability between applications and current and emerging security systems.

30 Another advantage of the invention is that it may employ and enhance the utility of the public key infrastructure (PKI) scheme. The invention turns long-lived digital certificates into ephemeral certificates (name assertions) therefore minimizing overall management and

overhead. Specifically, the invention eliminates the need for servers to check certificate revocation lists (CRLs), a process that has plagued the proliferation of PKI.

Another advantage of the invention is that it may employ and enhance the utility of Kerberos, where the client has had to communicate with the Kerberos authentication mechanism every time the client contacts a specific server for the first time. Name assertions, as used by the present invention, are general and can be used to prove identity to any server. Therefore, a single valid name assertion can be presented to any server. This eliminates the need for the client to contact the authenticating mechanism every time it wants to communicate with a different server.

Another advantage of the invention is that it may employ and enhance the utility of secure remote password (SRP). SRP enables authentication of a client to a server, but it does not strongly authenticate the server to the client. The invention improves on this by implementing a strong, mutual authentication protocol.

Another advantage of the invention is that it permits a graceful migration path, unlike existing authentication technologies which require all user and servers to be enabled with that technology and which result in an all- or-none proposition for the entire enterprise. Using the present invention, an organization can change its authentication mechanism without affecting its users or servers.

And another advantage of the invention is that it is lightweight, its architecture permits a very lightweight implementation making it suitable for a wide variety of deployment scenarios.

These and other objects and advantages of the present invention will become clear to those skilled in the art in view of the description of the best presently known mode of carrying out the invention and the industrial applicability of the preferred embodiment as described herein and as illustrated in the several figures of the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The purposes and advantages of the present invention will be apparent from the following detailed description in conjunction with the appended figures of drawings in which:

5 FIG. 1 is a block diagram depicting how the invention includes a set of cooperating components which execute in different administrative domains;

FIG. 2 is a block diagram depicting an example of how the invention may be applied by two companies for collaboration in the development of a new product;

10 FIG. 3 is a block diagram depicting an example of how the invention may be applied by an outside managed security services provider (MSSP) to provide authentication for customers; and

FIG. 4 is a block diagram that also depicts how the invention includes a number of interacting components, expanding on FIG. 1 by also showing various options and usage with multiple authentication mechanisms.

RECORDED DOCUMENT

BEST MODE FOR CARRYING OUT THE INVENTION

A preferred embodiment of the present invention is a Federated Authentication Service Technology (FAST). As illustrated in the various drawings herein, and particularly in the view of FIG. 1, a preferred embodiment of the invention is depicted by the general reference character 10. To assist in understanding the following discussion, a glossary is also provided after the Industrial Applicability section.

FIG. 1 depicts how FAST 10 includes a set of cooperating components which execute across different administrative domains in a network, with the boundaries depicted here with dashed lines. Thus, a subject domain 12 is stylistically depicted as bordering an agent domain 14, an authentication domain 16, and a server domain 18. The subject domain 12 includes a subject 20 (not necessarily human) and a client application or applet (client 22). The agent domain 14 includes an authentication agent 24, a mechanism resolution process 26, a mechanism repository 28, and a mechanism registration process 30. The authentication domain 16 includes an authentication mechanism 32 and a protocol proxy 34 (which may alternately reside in the authentication domain 16). The server domain 18 includes a server application 38.

From the subject domain 12 it is desired to obtain access to the server domain 18. Such access is predicated upon a successful authentication in the authentication domain 16, and the agent domain 14 facilitates the authentication. FIG. 1 and this description are simplified, somewhat, to present key points but, as will be described presently, in typical embodiments the FAST 10 will include a number of subjects 20, authentication mechanisms 32, and server applications 38, all employing the services of one authentication agent 24. When multiple entities of a similar type are present, they are each treated as respective domains.

Before describing FIG. 1 in more detail it will help to appreciate the role in FAST 10 of a core concept called "name assertion." A name assertion is a type of credential. More specifically, in actual implementation, it is a digitally signed data structure containing a declaration of identity which is presentable to establish a claimed identity.

FAST 10 preferably uses the secure remote password (SRP) protocol to authenticate an entity that presents a name assertion. In FAST 10 this is accomplished by including an SRP verifier in the name assertion, and then providing the SRP secret to the entity that must authenticate itself. Thus, for example, when a client 22 presents a name assertion to a server

application 38, the server application 38 retrieves the SRP verifier from the name assertion and uses the SRP scheme to challenge the client 22 to prove possession of the SRP secret.

Turning now to a detailed description of FIG. 1, it depicts the overall flow of authentication in FAST 10. A series of steps, steps 40-52, encompass the actual authentication process and, generally but not always, occur every time a subject 20 must be authenticated to a server application 38. In contrast, steps 56-58 encompass a registration process that only need occur when an authentication mechanism 32 registers or updates itself to the agent domain 14 (changes the mechanism repository 28).

In step 40 the subject 20, who must authenticate itself, uses the client 22 to initiate the process of obtaining access to the server application 38. It should be noted that while the client 22 here is shown as distinct from the subject 20, that need not be the case if the subject 20 can have the functionality of the client 22 integrated into it, say, if the subject 20 is non-human.

In step 42 the client 22 contacts the authentication agent 24 and passes to it the name of the subject 20 and their domain. The client 22 can, optionally, also send the name of a particular authentication mechanism 32 and other data (typically including method and strength of authentication which the authentication mechanism can provide, if pertinent). A successful interaction between the client 22 and the authentication agent 24 produces information about exactly one authentication mechanism 32 for the client 22 to use. Otherwise, the authentication agent 24 returns an error condition indicating that no authentication mechanism 32 matches the request by the client 22.

In step 44 the authentication agent 24 uses the mechanism resolution process 26 to determine an appropriate authentication mechanism 32 for the client 22. If there is more than one which is appropriate, the authentication agent 24 uses its protocol with the client 22 (step 42 above) to resolve to exactly one authentication mechanism 32.

In step 46 the mechanism resolution process 26 uses the mechanism repository 28 to retrieve information about appropriate authentication mechanisms 32. This information could then be passed back to the client 22 for final mechanism resolution.

In step 48 the client 22 communicates an authentication request for access to the server application 38 to the protocol proxy 34 using a standard secure protocol. It should be noted that this need not go via the authentication agent 24 or any part of the agent domain 14 (but that it may optionally do so, being passed-thru in a manner described presently).

In step 50 the protocol proxy 34 receives the authentication request from the client 22 and translates it into the native protocol of the authentication mechanism 32. Consequently, there can be one protocol proxy 34 for each type of authentication mechanism 32. The protocol proxy 34 next communicates the translated request to the authentication mechanism 32. Upon successful 5 authentication, the protocol proxy 34 receives back from the authentication mechanism 32 a response including attributes and access rights of the subject 20. The protocol proxy 34 then creates a name assertion and, optionally, entitlements. The protocol proxy 34 translates this into an authentication response which it transmits back to the client 22.

In step 52 the client 22 delivers the authentication response to the server application 38. 10 The client 22 and the server application 38 then engage in a protocol that proves the client 22 is the proper owner of the name assertion.

As noted, the above steps 40-52 generally (but not always) occur every time a subject 20 must authenticate itself to a server application 38. To the extent that name assertions are reusable, the subject 20 can present it to any server application 38, any number of times. To initially register an authentication mechanism 32 the following steps 56-58 are used.

In step 56 the authentication mechanism 32 contacts the mechanism registration process 30 to initiate registration.

In step 58 the mechanism registration process 30 enters information about the authentication mechanism 32 into the mechanism repository 28, and thereafter the steps 40-52 may be employed. The protocol proxy 34 may already be available to the authentication mechanism 32, or it can be provided or made available by the mechanism registration process 30 during these registration steps.

Each of the components of FAST 10 has a protocol which it employs when 25 communicating with the others within FAST 10. The subject 20 uses an environment-specific device to authenticate itself. FAST 10 is environment-agnostic, so a subject 20 may authenticate in any environment using any type of credential. Some examples of such credentials include user ID/passwords on a computer or a hand-held device, digital certificates and an associated private key, and biometric data such as a fingerprint or iris-scan.

The client 22 is an application or an applet that helps the subject 20 authenticate itself. 30 The clients 22 run in environment-specific platforms and interact with devices that produce the necessary credentials. For example, a client 22 may interact with a biometric device to gather

fingerprint data.

The subject 20 may obtain the client 22 in a number of different manners. For example, it may be pre-provisioned with the client 22 or it may download it "on the fly" from the authentication agent 24, the server application 38, or elsewhere.

5 As was noted above, the client 22 interacts with the authentication agent 24 in order to determine the most appropriate authentication mechanism 32. The authentication agent 24 can return more than one appropriate authentication mechanism 32. The client 22 therefore may have a callback mechanism to enable local determination of exactly which authentication mechanism 32 it should use. For example, the callback mechanism can interact with the subject 20 to 10 determine this or it may consult a configuration repository.

The authentication agent 24 brokers between the client 22 and the mechanism resolution process 26. The client 22 must minimally send the authentication agent 24 the name and the domain of the subject 20, but it may optionally also send other information. The authentication agent 24 uses this information to resolve to an appropriate authentication mechanism 32. The authentication agent 24 then returns information identifying authentication mechanisms 32 to the client 22. If more than one is supplied, the client 22 uses its callback mechanism to select exactly one.

20 A specific scenario provided for is when the client 22 chooses the authentication mechanism 32 without the help of the authentication agent 24. For example, the client 22, through its interaction with the subject 20 may choose an authentication mechanism 32. In this case the client 22 has two choices. First, the client 22 may send a request for the specific authentication mechanism 32 it desires to the authentication agent 24. Subsequently, the authentication agent 24 merely checks to ensure that the authentication mechanism 32 can indeed authenticate the subject 20. Second, the client 22 may directly contact the protocol proxy 34 to 25 start the authentication process with the authentication mechanism 32, and thus effectively bypass the mechanism resolution process 26.

The mechanism resolution process 26 is a "black-box" process. In the preferred embodiment, it receives an input document in XML format and produces a list of zero or more appropriate authentication mechanisms 32. The input document minimally consists of the name 30 of the subject 20 and their domain. The input document can be expanded to include any other data (e.g., an authentication strength) that can help choose the set of authentication mechanisms

32. For instance, consider a subject 20, John Doe, who has a work account at "A.com," and various other accounts with "B.com," "C.com," etc. Mr. Doe may need to access the resources of Z.com in his work capacity, and therefore needs to authenticate as "JohnDoe@A.com." This will require resolution to an appropriate authentication mechanism 32 for Z.com, for instance, may be 5 set up with a mask "*@A.com" to direct resource requests to use a particular authentication mechanism 32.

The mechanism repository 28 is a database that contains information about authentication mechanisms 32. It should be noted that a "mechanism" specifies exactly how to contact and work with the protocol proxy 34 of each authentication mechanism 32, e.g., what URL to use to reach 10 it, etc. The mechanism resolution process 26 and the mechanism registration process 30 use the mechanism repository 28 to resolve and to register the authentication mechanisms 32, respectively.

The mechanism registration process 30 is what the authentication mechanisms 32 use to register themselves or to modify information about themselves in the mechanism repository 28. As compared to authentication, the mechanism registration process 30 happens infrequently.

Each protocol proxy 34 mediates between its authentication mechanism 32 and the clients 22. In the preferred embodiment, the protocol proxies 34 use a standard security protocol expressed in XML to communicate with the clients 22, and a mechanism-specific protocol to communicate with their respective authentication mechanisms 32. At the end of a successful authentication, a protocol proxy 34 produces a signed document including a name assertion and, optionally, an entitlement. Examples are provided in Appendix A.

Each protocol proxy 34 must itself be authentic and have a valid name assertion. In this manner, the client 22 has to only reveal sensitive credentials to an authentic protocol proxy 34. Moreover, a server application 38 has to only trust name assertions that are produced by an 25 authentic protocol proxy 34. A protocol proxy 34 therefore uses its own name assertion to authenticate to a client 22 and to sign the name assertions and entitlements which it produces on behalf of its authentication mechanism 32. (This is described further, below.)

The authentication mechanism 32 is a process that authenticates a subject 20 according to a specific protocol. Note, however, the protocol proxy 34 hides the specific protocol of the 30 authentication mechanism 32 used for this from the subject 20 and the client 22. Thus, the authentication mechanism 32 may be entirely conventional, yet still be able to function with an

overall scheme of multiple authentication types and sources, that being a major benefit provided by the inventive FAST 10.

Each name assertion contains a SRP verifier. The rightful owner of the name assertion possesses the corresponding SRP secret. Using the SRP verifier and the SRP secret, any two parties can authenticate each other using the SRP protocol. In FAST 10 the SRP protocol is extended to enable mutual authentication between any two parties (e.g. client 22 and server application 38, client 22 and protocol proxy 34, etc.).

Each name assertion contains a set of public values. In FAST 10 the use of these public values is extended for signature verification. That is, the owner of the name assertion can use its SRP secret to authenticate itself and to produce a digital signature. The party to whom the owner presents the name assertion can then use it to authenticate the owner and to verify the digital signature of the owner.

Because the name assertion itself is signed by the protocol proxy 34 that produces it, the signature of the owner is bound to its authenticated identity. Additionally, because name assertions are ephemeral, there is no requirement for certificate revocation lists (CRL). Note that an authentication mechanism 32 that uses digital certificates to authenticate its subjects 20 can trivially set the name assertion validity period to coincide with the production of the next CRL. In this manner the subject 20 must re-authenticate itself on or before the publication of the next CRL.

The clients 22 need to trust the protocol proxies 34 (which represent the authentication mechanisms 32). This is especially true in the case of a protocol proxy 34 that receives secret credentials from a client 22. The protocol proxies 34 therefore authenticate themselves to a mechanism-authenticating mechanism. This type of authentication is identical to a client 22 authenticating with any authentication mechanism 32. The result is a name assertion that the protocol proxy 34 uses to engage in mutual authentication with the client 22 and to sign name assertions for the clients 22 it authenticates.

When a protocol proxy 34 authenticates with a mechanism-authenticating mechanism, it receives a name assertion in the same manner that any client 22 would. Such a name assertion is signed by the mechanism-authenticating mechanism (or, more precisely, by the protocol proxy 34 ahead of the mechanism-authenticating mechanism that mediates between the protocol proxy 34 needing to be authenticated, i.e., itself temporarily acting in the role of a client 22). A client

22 can verify the digital signature of the name assertion of the protocol proxy 34 to ascertain its authenticity, as provided by the mechanism-authenticating mechanism. This is similar to verifying a chain of digital certificates, except that in this case, name assertions and certificates are combined to provide a practical solution where all parties need not have digital certificates 5 (only the mechanism-authenticating mechanism would need a digital certificate).

Recall that the client 22 first establishes contact with the protocol proxy 34. In actual implementation it would be very beneficial if the client 22 can view the authentication agent 24 as the protocol proxy 34. This is, in fact, possible in FAST 10, with the reason for that being that the protocol proxy 34 and the client 22 can establish a mutually authenticated session whose 10 protocol data is completely hidden from any process that acts as a pass-thru. That is, the authentication agent 24 can act as a transparent protocol proxy 34. In this case, the authentication agent 24 acts as a pass-thru. This provides convenience and efficiency for the client 22, which behaves as if it is dealing with one server (the authentication agent 24), while preserving the overall security of the authentication process.

In FAST 10 authentication types and sources are abstracted. It is therefore possible to develop protocol proxies 34 that can facilitate the authentication of any subject 20 with any authentication mechanism 32. Four authentication types and their implementations in FAST 10 are now described.

The first implementation to consider uses a user ID and secret credentials. This is a variation on the most popular form of authentication employed today. Here, the user (subject 20) presents a secret credential to the protocol proxy 34. The secret credential could, for instance, be a password or biometric data.

FAST 10 implements this authentication type by establishing a secret link between the protocol proxy 34 and the client 22. This secret link is irrespective of any other communication 25 intermediaries. That is, a process that is acting as pass-thru between the client 22 and the protocol proxy 34 cannot discern the data.

The client 22 and the protocol proxy 34 establish the secure link as follows. First, the client initiates a dialog with the protocol proxy 34. Second, the protocol proxy 34 presents its own name assertion to the client 22. Third, the client 22 uses the verifier in the name assertion it 30 receives to authenticate the protocol proxy 34 via the SRP method. Fourth, the client 22 and the protocol proxy 34 establish a secured link via the SRP method.

Once the client 22 establishes the secured link with the protocol proxy 34, it delivers the credentials and, upon successful authentication, receives a name assertion from the protocol proxy 34.

5 The second implementation to consider uses verifier-based authentication. In this form of authentication the client 22 proves possession of a secret credential to the protocol proxy 34. The protocol proxy 34 has a verifier that matches the secret credentials of the client 22. In the preferred embodiment of FAST 10 this authentication type is implemented using SRP with mutual authentication (as discussed above).

The third implementation to consider uses digital certificates in a PKI scheme.

10 Authentication using digital certificates does not require a secret link. The protocol proxy 34 uses a standard challenge/response protocol to prove possession of private key by the client 22. Upon successful authentication, the client 22 receives a name assertion from the protocol proxy 34.

15 The fourth implementation to consider uses a previously issued name assertion. Each name assertion contains a verifier whose corresponding secret is maintained by the owner of the name assertion (i.e., the client 22). A client 22 who has a valid name assertion authenticates with the protocol proxy 34 using SRP with mutual authentication.

20 While the above generally summarizes the invention, FAST 10 may incorporate additional capabilities. One of these is strength of authentication. A client 22 can specify the strength with which it wants to authenticate itself. This data is an input to the mechanism resolution process 26. Conversely, each name assertion has a strength indicator, which indicates to a server application 38 the strength with which the subject 20 actually authenticated itself. The clients 22 can specify other environmental variables as part of the request to authenticate. These variables are input to the mechanism resolution process 26. The name assertions have an 25 expiration time, but a client 22 can renew its name assertion if it can prove ownership of it, and if the renewal count or time period in the original name assertion permits further renewal.

FAST 10 does not require special provisions for authenticating to multiple mechanisms. Instead, this capability is implemented by simply requiring that the client 22 authenticate with each authentication mechanism 32 and receive a separate name assertion.

30 Two examples of embodiments of FAST 10 are now provided. Each is a practical (but different) deployment scenario. Co-operating partners requiring cross-organization

authentication is the context of the first example.

FIG. 2 is a block diagram depicting the inventive FAST 10 in use by two companies collaborating with each other on the development of a new product, AliCo 112 and Zyland 114 (any similarity to actual businesses is purely coincidental). AliCo 112 and Zyland 114 are 5 represented as regions lying on respective sides of an administrative control boundary 116. Both companies have employees, stylistically shown respectively as an a-employee 118 and a z-employee 120 (instances of the subject 20 of FIG. 1; the presence of clients, etc. is also implicit here, since humans cannot directly access information systems). These employees 118, 120 collectively form a project team 122.

10 For this example, we presume that the relationship between AliCo 112 and Zyland 114 is not a permanent one, but that they do have an extremely tight product development cycle. Accordingly, for the duration of the project the members of the project team 122 require access to the information system of their partner company. That is, the a-employee 118 and the z-employee 120 need to use the tools of both companies, stylistically shown as an a-application 124 and a z-application 126.

15 AliCo 112 has an a-authentication mechanism 128 and Zyland 114 has a z-authentication mechanism 130 (instances of the authentication mechanism 32 of FIG. 1). These authenticate the company's respective users (employees 118, 120 and potentially many others not pertinent to this example). The a-authentication mechanism 128 used by AliCo 112 is an LDAP directory with user IDs and passwords. In contrast, the z-authentication mechanism 130 used by Zyland 114 is digital certificates and associated certificate revocation lists (CRL). Additionally, both AliCo 112 and Zyland 114 use their own internal systems to manage the entitlements of their respective employees.

20 Existing solutions to this problem would require one or more of the following: the members of the project team 122 individually registering with the authentication mechanism 128, 130 of the partner company; the members of the project team 122 having a set of entitlements in the repository of the partner company; and the software tools (applications 124, 126) in both companies understanding the authentication protocol of the partner company (e.g., the applications must be PKI-enabled or Kerberized).

25 This quickly becomes unwieldy. When one company revokes the credentials or changes the entitlements of an employee it must promptly inform its partner. The partner must then

promptly reflect that change in its own repository. Changing the software tools, either by outright addition or by upgrade, also may be effected.

Allegedly simpler solutions would require each member of the project team 122 to obtain a digital certificate from a commonly trusted certificate authority (CA). However, even if digital certificates could be obtained and managed easily, such certificates cannot practically store entitlement information. For example, any modifications to the entitlement would invalidate the certificate. This is why there are long-lived identity certificates and short-lived attribute certificates in the PKI scheme, and why more than 99% of all certificates in use today are identity certificates.

Now consider the solution using FAST 10, as depicted in FIG. 2. AliCo 112 and Zyland 114 would deploy small software protocol proxies 132, 134 (i.e., instances of the protocol proxy 34 of FIG. 1) at their authentication source, and a standard XML adapter 136, 138 in front of their applications 124, 126 (for this example we presume that the applications 124, 126 are not able to directly handle name assertions and need the adapter 136, 138 for this). Immediately thereafter, the a-application 124 of AliCo 112 will recognize the z-employee 120 of Zyland 114, and can determine his or her entitlements. And the reverse is also true. If one company revokes the credentials or changes the entitlements of one its users, the other company will know it as soon as the next authentication attempt occurs.

Before closing with FIG. 2, it should be noted that it depicts a simple embodiment of the inventive FAST 10, in that no equivalent of the agent domain 14 and its components is depicted. Since the employees 118, 120 on the project team 122 here are only accessing the applications 124, 126 of the partner company, they will easily know the respective authentication mechanism 128, 130 need and not require the assistance of an agent in resolving one.

Many organizations today are outsourcing their security services to outside managed security services providers (MSSPs). The challenge of MSSPs then is to streamline their operation and realize the economy of scales. This provides the context of the next example.

FIG. 3 is a block diagram depicting the inventive FAST 10 providing authentication for a MSSP 210. In this second example, the MSSP 210 needs to support several types of authentication types across several hundreds of customers 212 (entities including instances of the subject 20 of FIG. 1). While the MSSP 210 wants to manage the authentication process, it does not want to be the source of authentication. Nor does it want to be in a position to see sensitive

credentials (e.g., passwords). This provides a number of benefits to the MSSP 210 and its customers 212. The MSSP 210 can support any customer 212, even those with extremely high security requirements. It can also provide an incremental solution, migrating customers 212 from a minimally managed to a totally managed solution. Not having to see credentials also reduces 5 the legal liabilities of the MSSP 210 arising from any security breaches. It also helps the MSSP 210 to avoid costly implementations of chain-of-trust rules.

Legacy solutions to this problem require that the MSSP 210 set up a completely separate and trusted system for each customer 212. The MSSP 210 must then protect each system with the rigor that meets the demands and expectations of each customer 212. Most importantly, the 10 solution is all-or-none, inhibiting an incremental deployment.

Now consider the solution with FAST 10, as depicted in FIG. 2. The MSSP 210 can deploy a single authentication engine 214 for managing authentication for all of its customers 212. The actual authentication source can be at the customer 212, at the MSSP 210, or at a third place (e.g., the authentication domains 216 shown). In fact, the authentication source can move its location and change its administrative authority (from the customer 212 to the MSSP 210 or vice versa) with no effort at all.

In the example in FIG. 3, the authentication sources (paired instances of the protocol proxy 34 and the authentication mechanism 32 of FIG. 1) are depicted as being at a third place. This arrangement has deliberately been used in FIG. 1 and FIG. 3 to emphasize the fact, and the 20 ability of FAST 10 to accommodate that fact, that the authentication sources may be quite removed from the users and tools (instances of the subject 20 and server application 38 of FIG. 1).

Turning now to FIG. 4, a summary of FAST 10 is now discussed. The word "federated" has two specific meanings herein: a system is federated if it supports multiple authentication 25 types, and multiple authentication sources. Prior art approaches address multiple authentication types but largely ignore multiple authentication sources. Thus, the resulting products can support many authentication types for a single organization but cannot support inter-organization authentication. In this discussion we use the word "mechanism" to refer to a specific authentication type at a specific authentication source.

30 FIG. 4 is a block diagram that again depicts how FAST 10 includes a number of interacting components, extending somewhat on FIG. 1. A subject 312 is the entity that needs to

authenticate itself (user, device, etc.; again the presence of a client is implicit here, and one may even be integrated into a non-human subject 312). A server application 314 provides service to the subject 312. In order to do so the server application 314 must know the authenticated identity of the subject 312, and possibly the entitlements of the subject 312. A mechanism registration 5 module 316 performs the process that binds the authentication mechanisms 318a, 318b, 318c to the subjects 312 (e.g., all users at alico.com must authenticate against ldapserver.alico.com). A mechanism repository 320 is the database that holds information about mechanisms such as their location, type, credentials (e.g., a digital certificate), and protocol. A mechanism resolution module 322 performs the process that resolves the name of a subject 312 to one or more 10 mechanisms. An authentication agent 324 is the process that finds the proper authentication mechanism and facilitates the authentication protocol between the subject 312 and a protocol proxy 326a, 326b, 326c. The protocol proxies 326a, 326b, 326c are the interface between the authentication agent 324 and the authentication mechanisms 318a, 318b, 318c.

There is one protocol proxy per specific authentication type (e.g., and LDAP user ID/Password proxy). The protocol proxy can co-reside with either the authentication mechanism or the authentication agent, resulting in possible different deployment scenarios. An authentication mechanism is the specific mechanism, embodying the authentication type and location of authentication. In FIG. 4 the protocol proxies 326a, 326c reside, respectively, with the authentication mechanisms 318a, 318c; and the protocol proxy 326b resides with the authentication agent 324.

If a subject 312 wants to use a server application 314, the process begins with the subject 312 contacting the authentication agent 324. The authentication agent 324 uses the mechanism resolution module 322 to process and resolve the name of a subject 312 (e.g., james@alico.com) to an authentication mechanism 318a (for instance). The authentication agent 324 then uses the 25 protocol proxy 326a (used as an example now) to facilitate passing of credentials between the subject 312 and the authentication mechanism 318a. Note that the authentication agent 324 does not see any sensitive credentials that the subject 312 passes to the protocol proxy 326a (e.g., passwords). The subject 312 and the protocol proxy 326a each establish a secure tunnel. Thus, for mechanisms that require knowledge of sensitive credentials it is best to co-locate the proxy at 30 the authentication mechanism, as is shown in FIG. 4 for the authentication mechanism 318a and protocol proxy 326a.

The subject **312** and the protocol proxy **326a** engage in the process of authentication. If the subject **312** is authentic, then the protocol proxy **326a** produces a digitally signed document consisting of a name assertion and entitlements. In the inventors' presently preferred embodiment, the actual format of the document is according to the security services markup language (S2ML), which is a draft XML standard for communicating security information. Once the subject **312** receives the S2ML document it can pass it to any application that understands the simple S2ML elements and attributes. If necessary, for any application (e.g., the server application **314** in FIG. 4) that does not understand S2ML, an XML adapter **328** can be used to provide this capability.

10 The power of **FAST 10** lies in its simplicity and in its inherent security. Components of **FAST 10** can be deployed in a number of ways, yielding deployment scenarios that simultaneously meet security and business requirements of organizations.

20 The mechanism resolution module carries out a process which takes into account the environment within which the subject is operating. For example, asking the subject for credentials through a hand held device would be more different than obtaining such with a retina-scanning device. Thus, a subject can authenticate in whatever manner desired, leading to a more positive experience.

The authentication mechanisms need not be modified. Instead, the protocol proxies interface between the authentication agent and the authentication mechanisms. As a result, organizations can quickly leverage their existing authentication mechanisms, leading to cost savings and protection of their investments.

Authentication is decoupled from the applications. This permits organizations to change their underlying authentication mechanisms, either incrementally or all at once, without ever needing to modify their own or third party applications used by their own subjects.

25 The authentication mechanisms run at the location that owns them (i.e., within their boundary of administrative control). The exchange of information between an authentication mechanism and a subject is private. Therefore, the authentication agent can never see or steal sensitive credentials.

30 The subjects need not be just human users. They can be applications, devices, processes, etc. As a result, **FAST 10** is applicable in multiple environments, involving human and non-human subjects.

Because the components of **FAST 10** can run anywhere, and because these components can be under different domains of control, it is imperative that each individual component be secure, both internally and in its communication with other components. This total security of **FAST 10** is now discussed.

5 With continued reference to FIG. 4, the authentication agent 324 mediates authentication requests and responses between the subject 312 and the protocol proxy 326a. However, in no case can the authentication agent 324 view secret credentials. The reason for this is that the subject 312 and the protocol proxy 326a set up a secret key that is not known to the authentication agent 324. Additionally, the authentication agent 324 cannot modify the result of 10 the authentication (the S2ML name assertion and entitlement). The reason for this is that the protocol proxy 326a digitally signs the authentication response.

In the example of the FAST 10 depicted in FIG. 4, a dashed line depicts the boundary of the agent domain 330 and it can be seen that all communication between the subject 312 and the protocol proxies 326a, 326b, 326c passes through the agent domain 330. Nonetheless, even this arrangement is secure for the reasons just described.

The protocol proxy establishes a secure protocol with the subject **312**. The protocol proxy can run co-resident with the authentication agent **324** (as protocol proxy **326b** does in FIG. 4), or co-resident with an authentication mechanism (as protocol proxy **326a** does with the authentication mechanism **318a** in FIG. 4). If the subject **312** is supplying sensitive credentials, it is best for the protocol proxy to run co-resident with the authentication agent **324**.

The protocol proxies produce signed S2ML name assertions and entitlements. The subject 312 can use the S2ML document to authenticate to any server (e.g., the server application 314). In this respect, there is an important difference between FAST 10 and other authentication architectures. In other authentication architectures (e.g., Kerberos) the name assertion is targeted for a specific server. As such, the server cannot steal the name assertion. However, the subject in such a scheme would need a different name assertion for every server.

In order to protect the S2ML document from being replayed or stolen, the protocol proxy delivers a SRP secret to the subject, and includes a SRP verifier for the server application in the S2ML document. In this manner the server can always verify that the subject was the original and intended recipient of the S2ML document, but the server can never use the S2ML document to pose as the subject. In summary, the S2ML name assertion and entitlement is analogous to a

digital certificate that the application server can only use to authenticate the subject.

The authentication mechanisms do not directly communicate with any other component of **FAST 10**. Instead, the protocol proxies are the interface between the authentication mechanisms and the other components of **FAST 10**. Thus, the communication between a protocol proxy and an authentication mechanism can be as secure as desired.

FAST 10 may employ widely used and trusted security industry standards. For instance, X.509 version 3 digital certificates may be used. The components of **FAST 10** can use such digital certificates as the basis for SSL/TLS connections and for digital signature verification. As noted above, security services markup language (S2ML) may also be used. The protocol proxies can produce XML documents that comply with S2ML schema. XML digital signatures are another standard which may be used. **FAST 10** may use XML-signature specifications for production of digital signatures in S2ML name assertions and entitlements. The use of the secure remote password (SRP) standard permits **FAST** to use the SRP authentication and key exchange system, as specified in RFC 2945. Furthermore, the enhanced version of SRP described herein provides additional benefits. **FAST 10** may also use various standard encryption and message digest algorithms for protecting the privacy and integrity of its protocol data. These may include Diffie-Hellman, RSA, AES, SHA-1, and keyed-hashing for message authentication, as defined in RFC 2104.

The ability to share information across enterprise boundaries enables organizations to create inter-enterprise business and to gain a competitive advantage. Simultaneously, information constitutes an important corporate asset and must be protected commensurate with its value. In order to protect information, organizations need to ascertain the identity of users.

In the real world each one of us is known and authenticated in many ways. Our friends and families know us by our voices or likeness, financial and legal institutions know us by our signatures and officially issued identifications, and law enforcement institutions know us by our fingerprints.

Global single sign-on does not mean a single source of authentication, nor does it mean a single type of authentication. It is the effective use of multiple, federated authentication sources that ultimately leads to global single sign-on.

FAST 10 is the only technology today that permits use of multiple authentication types and multiple authentication sources from different domains of control. By using **FAST 10**,

organizations can satisfy the most stringent security requirements while leveraging their existing information systems to quickly implement business relationships.

While various embodiments have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the invention should not be limited by any of the above described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

100-2292860

INDUSTRIAL APPLICABILITY

The present **FAST 10** is well suited for use to authenticate subjects 20, 118, 120, 212, 312 to server applications 38, 124, 126, 314. As shown in FIG. 2, this can be on a small scale, for just 5 a few subjects and server applications, or this can be on a very large scale, for potentially thousands, as depicted in FIG. 3. These are current needs which **FAST 10** well serves, and ones which **FAST 10** may be immediately implemented to serve.

10 **FAST 10** permits the use of multiple authentication types and multiple authentication sources from different domains of control, regardless of how disharmonious these may be. **FAST 10** simply abstracts both authentication type and source. Accordingly, organizations can satisfy the most stringent security requirements, choosing security infrastructure as they wish or leveraging their existing ones to quickly implement business relationships.

The ability to support multiple types of authentication permits organizations to deploy one authentication type and migrate to other types as desired, say, as their needs grow or as more robust types emerge. Existing authentication technologies, such as Kerberos or public key infrastructure, require all users and servers to be enabled with that technology. The result is an all-or-none proposition for the entire enterprise. Using **FAST 10**, an organization can change its authentication mechanism without affecting its users or servers.

The ability to support multiple authentication sources permits users and service providers to choose where to authenticate. For example, in order to pay a bill a user would have to authenticate with his bank. However, in order to view medical records a user would have to authenticate with her healthcare provider, or perhaps her employer.

25 **FAST 10** permits production and verification of signatures using name assertions. This eliminates the need to use digital certificates for production and verification of signatures, which improves the speed and efficiency of signature production and verification. Name assertions may be used as a basis to renew an existing name assertion. This eliminates the need to use digital certificates every time authentication is needed, which improves on the speed and efficiency of the authentication process.

30 As has been described, **FAST 10** may employ and enhance the utility of Kerberos, the public key infrastructure (PKI) scheme, or particularly the secure remote password (SRP) protocol, improving upon it by implementing a strong, mutual authentication protocol.

FAST 10 is inherently secure. It permits a hierarchy of trust wherein authenticating mechanisms must authenticate themselves. In this manner, a subject need only reveal sensitive credentials to and only trust assertions of an authentic mechanisms. FAST 10 also permits flexible credential expiration, overcoming limitations of prior architectures, which are rigid about who sets the expiration time of credentials. FAST 10 permits credential expiration to be requested by the client, the server, or the mechanism.

For the above, and other, reasons, it is expected that the FAST 10 of the present invention will have widespread industrial applicability. Therefore, it is expected that the commercial utility of the present invention will be extensive and long lasting.

2025 MAR 22 2026 10
U.S. PATENT AND TRADEMARK OFFICE

GLOSSARY

Authentication Agent:

A process that facilitates authentication between a Subject and an Authentication Mechanism. The Authentication Agent itself never authenticates a Subject.

Authentication Mechanism:

A process that authenticates a Subject according to a specific protocol.

Administrative Domain:

The set of devices, people and processes under the control of the same entity.

Boundary of Administrative Control:

The boundary between Administrative Domains.

Client:

A process that a Subject uses to authenticate itself.

Client applet:

A specific implementation of a Client where the authentication code can be downloaded dynamically, or reside locally on the client, and run in a browser.

Client application:

A specific implementation of a Client where the authentication code resides on the client and runs as a stand-alone process.

Credentials:

Data that is presented to establish claimed identity.

Entitlement:

A data structure that contains access decision information. A Server uses Entitlements to determine what a Subject can do.

Mechanism-Authenticating; Mechanism:

An Authentication Mechanism that can authenticate other Authentication Mechanisms as its Subjects.

Mechanism Registration:

A process that an Authentication Mechanism uses to register itself in the Mechanism Repository.

Mechanism Repository:

A repository containing information about Authentication Mechanisms. The information includes type of mechanism, its protocol, its strength, and how a Client can contact it.

Mechanism Resolution:

The process that maps certain information about a Subject (e.g., name and Realm) to a set of Authentication Mechanisms.

5

Name Assertion:

A signed data structure containing a declaration of identity. A Name Assertion is presented to establish a claimed identity. A Name Assertion is a type of Credential.

Protocol Proxy:

10

A process that mediates between two other processes which do not understand each other's protocol.

Domain:

卷之三

A realm of authentication authority. An Authentication Mechanism can authenticate Subjects in one or more domains.

Server:

A process that provides service to a Subject through a Client. Servers require Subjects to authenticate themselves. In some cases Servers must also authenticate themselves to the Client/Subject.

Server Application:

20

An application that implements the functionality of a Server process. Usually, Server and Server Application are synonymous.

Strength:

An indication of the rigor of authentication.

Subject:

25

A user, application, device, process or any other entity that requires Authentication.

APPENDIX A

1. This is an example of an authentication request going from the client 22 to the protocol proxy 34.

5 <AuthRequest xmlns="http://ns.s2ml.org/s2ml">
 <ID>urn:PasswordAuthenticatorApplet:e58bd988ee:1</ID>
 <Date>2001-03-29T11:23:773-08:00</Date>
 <Credentials>
 <SecureLogin xmlns="http://sigaba.com/2000/12/sigabanet/fast">
10 <Name>logan@sigaba.com</Name>
 <Realm>SIGABA.COM</Realm>
 <A>
 BV+2pKUEf0i1I57/TIXlRzSkbIi7+lhWhSOuhglBoA0cAo6FxM111RWIS
G9iB0EEUPC+pplhTmzK1OsbVMs/TWGcBPuJVGX5lzhD3Far7Ozx3cIUU50AjGdr+HVwBt
4KJz4E8NrSZKs5fHBoefY3ykaGcKBezeyo5KN+Xx7mWQ=

 </SecureLogin>
 </Credentials>
 </AuthRequest>

20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995
1000
1005
1010
1015
1020
1025
1030
1035
1040
1045
1050
1055
1060
1065
1070
1075
1080
1085
1090
1095
1100
1105
1110
1115
1120
1125
1130
1135
1140
1145
1150
1155
1160
1165
1170
1175
1180
1185
1190
1195
1200
1205
1210
1215
1220
1225
1230
1235
1240
1245
1250
1255
1260
1265
1270
1275
1280
1285
1290
1295
1300
1305
1310
1315
1320
1325
1330
1335
1340
1345
1350
1355
1360
1365
1370
1375
1380
1385
1390
1395
1400
1405
1410
1415
1420
1425
1430
1435
1440
1445
1450
1455
1460
1465
1470
1475
1480
1485
1490
1495
1500
1505
1510
1515
1520
1525
1530
1535
1540
1545
1550
1555
1560
1565
1570
1575
1580
1585
1590
1595
1600
1605
1610
1615
1620
1625
1630
1635
1640
1645
1650
1655
1660
1665
1670
1675
1680
1685
1690
1695
1700
1705
1710
1715
1720
1725
1730
1735
1740
1745
1750
1755
1760
1765
1770
1775
1780
1785
1790
1795
1800
1805
1810
1815
1820
1825
1830
1835
1840
1845
1850
1855
1860
1865
1870
1875
1880
1885
1890
1895
1900
1905
1910
1915
1920
1925
1930
1935
1940
1945
1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
2025
2030
2035
2040
2045
2050
2055
2060
2065
2070
2075
2080
2085
2090
2095
2100
2105
2110
2115
2120
2125
2130
2135
2140
2145
2150
2155
2160
2165
2170
2175
2180
2185
2190
2195
2200
2205
2210
2215
2220
2225
2230
2235
2240
2245
2250
2255
2260
2265
2270
2275
2280
2285
2290
2295
2300
2305
2310
2315
2320
2325
2330
2335
2340
2345
2350
2355
2360
2365
2370
2375
2380
2385
2390
2395
2400
2405
2410
2415
2420
2425
2430
2435
2440
2445
2450
2455
2460
2465
2470
2475
2480
2485
2490
2495
2500
2505
2510
2515
2520
2525
2530
2535
2540
2545
2550
2555
2560
2565
2570
2575
2580
2585
2590
2595
2600
2605
2610
2615
2620
2625
2630
2635
2640
2645
2650
2655
2660
2665
2670
2675
2680
2685
2690
2695
2700
2705
2710
2715
2720
2725
2730
2735
2740
2745
2750
2755
2760
2765
2770
2775
2780
2785
2790
2795
2800
2805
2810
2815
2820
2825
2830
2835
2840
2845
2850
2855
2860
2865
2870
2875
2880
2885
2890
2895
2900
2905
2910
2915
2920
2925
2930
2935
2940
2945
2950
2955
2960
2965
2970
2975
2980
2985
2990
2995
3000
3005
3010
3015
3020
3025
3030
3035
3040
3045
3050
3055
3060
3065
3070
3075
3080
3085
3090
3095
3100
3105
3110
3115
3120
3125
3130
3135
3140
3145
3150
3155
3160
3165
3170
3175
3180
3185
3190
3195
3200
3205
3210
3215
3220
3225
3230
3235
3240
3245
3250
3255
3260
3265
3270
3275
3280
3285
3290
3295
3300
3305
3310
3315
3320
3325
3330
3335
3340
3345
3350
3355
3360
3365
3370
3375
3380
3385
3390
3395
3400
3405
3410
3415
3420
3425
3430
3435
3440
3445
3450
3455
3460
3465
3470
3475
3480
3485
3490
3495
3500
3505
3510
3515
3520
3525
3530
3535
3540
3545
3550
3555
3560
3565
3570
3575
3580
3585
3590
3595
3600
3605
3610
3615
3620
3625
3630
3635
3640
3645
3650
3655
3660
3665
3670
3675
3680
3685
3690
3695
3700
3705
3710
3715
3720
3725
3730
3735
3740
3745
3750
3755
3760
3765
3770
3775
3780
3785
3790
3795
3800
3805
3810
3815
3820
3825
3830
3835
3840
3845
3850
3855
3860
3865
3870
3875
3880
3885
3890
3895
3900
3905
3910
3915
3920
3925
3930
3935
3940
3945
3950
3955
3960
3965
3970
3975
3980
3985
3990
3995
4000
4005
4010
4015
4020
4025
4030
4035
4040
4045
4050
4055
4060
4065
4070
4075
4080
4085
4090
4095
4100
4105
4110
4115
4120
4125
4130
4135
4140
4145
4150
4155
4160
4165
4170
4175
4180
4185
4190
4195
4200
4205
4210
4215
4220
4225
4230
4235
4240
4245
4250
4255
4260
4265
4270
4275
4280
4285
4290
4295
4300
4305
4310
4315
4320
4325
4330
4335
4340
4345
4350
4355
4360
4365
4370
4375
4380
4385
4390
4395
4400
4405
4410
4415
4420
4425
4430
4435
4440
4445
4450
4455
4460
4465
4470
4475
4480
4485
4490
4495
4500
4505
4510
4515
4520
4525
4530
4535
4540
4545
4550
4555
4560
4565
4570
4575
4580
4585
4590
4595
4600
4605
4610
4615
4620
4625
4630
4635
4640
4645
4650
4655
4660
4665
4670
4675
4680
4685
4690
4695
4700
4705
4710
4715
4720
4725
4730
4735
4740
4745
4750
4755
4760
4765
4770
4775
4780
4785
4790
4795
4800
4805
4810
4815
4820
4825
4830
4835
4840
4845
4850
4855
4860
4865
4870
4875
4880
4885
4890
4895
4900
4905
4910
4915
4920
4925
4930
4935
4940
4945
4950
4955
4960
4965
4970
4975
4980
4985
4990
4995
5000
5005
5010
5015
5020
5025
5030
5035
5040
5045
5050
5055
5060
5065
5070
5075
5080
5085
5090
5095
5100
5105
5110
5115
5120
5125
5130
5135
5140
5145
5150
5155
5160
5165
5170
5175
5180
5185
5190
5195
5200
5205
5210
5215
5220
5225
5230
5235
5240
5245
5250
5255
5260
5265
5270
5275
5280
5285
5290
5295
5300
5305
5310
5315
5320
5325
5330
5335
5340
5345
5350
5355
5360
5365
5370
5375
5380
5385
5390
5395
5400
5405
5410
5415
5420
5425
5430
5435
5440
5445
5450
5455
5460
5465
5470
5475
5480
5485
5490
5495
5500
5505
5510
5515
5520
5525
5530
5535
5540
5545
5550
5555
5560
5565
5570
5575
5580
5585
5590
5595
5600
5605
5610
5615
5620
5625
5630
5635
5640
5645
5650
5655
5660
5665
5670
5675
5680
5685
5690
5695
5700
5705
5710
5715
5720
5725
5730
5735
5740
5745
5750
5755
5760
5765
5770
5775
5780
5785
5790
5795
5800
5805
5810
5815
5820
5825
5830
5835
5840
5845
5850
5855
5860
5865
5870
5875
5880
5885
5890
5895
5900
5905
5910
5915
5920
5925
5930
5935
5940
5945
5950
5955
5960
5965
5970
5975
5980
5985
5990
5995
6000
6005
6010
6015
6020
6025
6030
6035
6040
6045
6050
6055
6060
6065
6070
6075
6080
6085
6090
6095
6100
6105
6110
6115
6120
6125
6130
6135
6140
6145
6150
6155
6160
6165
6170
6175
6180
6185
6190
6195
6200
6205
6210
6215
6220
6225
6230
6235
6240
6245
6250
6255
6260
6265
6270
6275
6280
6285
6290
6295
6300
6305
6310
6315
6320
6325
6330
6335
6340
6345
6350
6355
6360
6365
6370
6375
6380
6385
6390
6395
6400
6405
6410
6415
6420
6425
6430
6435
6440
6445
6450
6455
6460
6465
6470
6475
6480
6485
6490
6495
6500
6505
6510
6515
6520
6525
6530
6535
6540
6545
6550
6555
6560
6565
6570
6575
6580
6585
6590
6595
6600
6605
6610
6615
6620
6625
6630
6635
6640
6645
6650
6655
6660
6665
6670
6675
6680
6685
6690
6695
6700
6705
6710
6715
6720
6725
6730
6735
6740
6745
6750
6755
6760
6765
6770
6775
6780
6785
6790
6795
6800
6805
6810
6815
6820
6825
6830
6835
6840
6845
6850
6855
6860
6865
6870
6875
6880
6885
6890
6895
6900
6905
6910
6915
6920
6925
6930
6935
6940
6945
6950
6955
6960
6965
6970
6975
6980
6985
6990
6995
7000
7005
7010
7015
7020
7025
7030
7035
7040
7045
7050
7055
7060
7065
7070
7075
7080
7085
7090
7095
7100
7105
7110
7115
7120
7125
7130
7135
7140
7145
7150
7155
7160
7165
7170
7175
7180
7185
7190
7195
7200
7205
7210
7215
7220
7225
7230
7235
7240
7245
7250
7255
7260
7265
7270
7275
7280
7285
7290
7295
7300
7305
7310
7315
7320
7325
7330
7335
7340
7345
7350
7355
7360
7365
7370
7375
7380
7385
7390
7395
7400
7405
7410
7415
7420
7425
7430
7435
7440
7445
7450
7455
7460
7465
7470
7475
7480
7485
7490
7495
7500
7505
7510
7515
7520
7525
7530
7535
7540
7545
7550
7555
7560
7565
7570
7575
7580
7585
7590
7595
7600
7605
7610
7615
7620
7625
7630
7635
7640
7645
7650
7655
7660
7665
7670
7675
7680
7685
7690
7695
7700
7705
7710
7715
7720
7725
7730
7735
7740
7745
7750
7755
7760
7765
7770
7775
7780
7785
7790
7795
7800
7805
7810
7815
7820
7825
7830
7835
7840
7845
7850
7855
7860
7865
7870
7875
7880
7885
7890
7895
7900
7905
7910
7915
7920
7925
7930
7935
7940
7945
7950
7955
7960
7965
7970
7975
7980
7985
7990
7995
8000
8005
8010
8015
8020
8025
8030
8035
8040
8045
8050
8055
8060
8065
8070
8075
8080
8085
8090
8095
8100
8105
8110
8115
8120
8125
8130
8135
8140
8145
8150
8155
8160
8165
8170
8175
8180
8185
8190
8195
8200
8205
8210
8215
8220
8225
8230
8235
8240
8245
8250
8255
8260
8265
8270
8275
8280
8285
8290
8295
8300
8305
8310
8315
8320
8325
8330
8335
8340
8345
8350
8355
8360
8365
8370
8375
8380
8385
8390
8395
8400
8405
8410
8415
8420
8425
8430
8435
8440
8445
8450
8455
8460
8465
8470
8475
8480
8485
8490
8495
8500
8505
8510
8515
8520
8525
8530
8535
8540
8545
8550
8555
8560
8565
8570
8575
8580
8585
8590
8595
8600
8605
8610
8615
8620
8625
8630
8635
8640
8645
8650
8655
8660
8665
8670
8675
8680
8685
8690
8695
8700
8705
8710
8715
8720
8725
8730
8735
8740
8745
8750
8755
8760
8765
8770
8775
8780
8785
8790
8795
8800
8805
8810
8815
8820
8825
8830
8835
8840
8845
8850
8855
8860
8865
8870
8875
8880
8885
8890
8895
8900
8905
8910
8915
8920
8925
8930
8935
8940
8945
8950
8955
8960
8965
8970
8975
8980
8985
8990
8995
9000
9005
9010
9015
9020
9025
9030
9035
9040
9045
9050
9055
9060
9065
9070
9075
9080
9085
9090
9095
9100
9105
9110
9115
9120
9125
9130
9135
9140
9145
9150
9155
9160
9165
9170
9175
9180
918

5

ID>

<Issuer>mechanism/AuthenticationServlet:e58bd98307@SIGABA.COM</Iss
uer>

10

<Date>2001-03-29T11:23:840-08:00</Date>

<Audiences>urn:*</Audiences>

<ValidityInterval>

<NotBefore>2001-03-29T10:23:840-08:00</NotBefore>

<NotAfter>2001-03-30T11:23:840-08:00</NotAfter>

</ValidityInterval>

<AuthData>

<AuthType>Login</AuthType>

COM</UserHandle>

<IdentityToken>

<dh-public-value xmlns="http://sigaba.com/2001/1/common/security">

<BigInteger>AOmstRCliTNQ8sbRHlrhVxcBK6SAYWdlMMIyR6MJKr
SiU/qhZcfYTWSGb0ni9MBDsnkdVBsa+/DA6PpjI45YhsCLi/ZExOMbqKSwLx2wp84Hu+s5S
a9XA+yTcA/WgzRTF+hjErU1fSImgYWd7326zA01D/WMhubYh+XM7nYgKv</BigInteger>

25

<BigInteger>Ag==</BigInteger>

<int>128</int>

<BigInteger>Jfb2Mli982CLj9sw0/0xgFVGpoICAIJejFE+VCTwymT0T
K4HunAtWcSs0PXFYv1agUlye0arQ+8OX0nKPV5dQjQP3oyuFVybeYHBSPQQD5RTY0Pk
FW/EU8iqcxNQcjvMYSu+oT9f60+t78B61vkQvE08c1NoH5UeoqW8Yyuig=</BigInteger>

30

</dh-public-value>

</IdentityToken>

</AuthData>

2000-2001-2002-2003-2004-2005-2006-2007-2008-2009-2010-2011-2012-2013-2014-2015-2016-2017-2018-2019-2020

5

10

20

25

30

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod
      algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"></CanonicalizationMethod>
    <SignatureMethod
      algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"></SignatureMethod>
      <Reference uri="#xp-pointer(..../..)"/>
      <Transforms>
        <Transform algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"/></Transform>
      </Transforms>
    <DigestMethod
      algorithm="http://www.w3.org/2000/09/xmldsig#sha1">
      <DigestValue>X3ZywSCxXHQKP6v9+/6r62/LGHc=</DigestValue>
    </DigestMethod>
    <Reference/></SignedInfo>
    <SignatureValue>
      RecJwLPgKpdMGMXlpoi0X8RPGgiQiH/OdzznUMGEpF/eRvb5I1ij
      mQ==</SignatureValue>
    <KeyInfo>
      <KeyValue>
        <DSAKeyValue>
          <P>AOmstRClilTNQ8sbRHlrhVxcBK6SAYWdlMMIyR6MJ
          KrSiU/qhZcfYTWSGb0ni9MBDsnkdVBsa+/DA6PpjI45YhsCLi/ZExOMbqKSwLx2wp84Hu+s
          5Sa9XA+yTcA/WgzRTF+hjErU1fSImgYWd7326zA01D/WMhubYh+XM7nYgKv</P>
          <Q>ANnezYYG5JOCcSNre+pJztsNA2n9</Q>
          <G>AKWsWEe8wT1KkMli+u05wZhODK4U0ZnutBGSY+4
        </DSAKeyValue>
      </KeyValue>
    </KeyInfo>
  </Signature>
```

<Y>BPT2HnTOBMfvFZb8UDn6fQ19gFkOsjGGZGGBRRhw
8Kyr06espL34MEavYnwwGW6D1VWbvIuaDHtUfRv6znkwzg0iCtsSnQ2FLD+rpsEQHok7pz
+UWDV4L5u4mhqzWWX3EiJmniOoxQduxqHjsbXm3XMqByWmeJOCVRIVeNa7Msw=</Y
>

5 </DSAKeyValue>
10 </KeyValue>
15 </KeyInfo>
20 </Signature>
25 </NameAssertion>
30 U7X7D6daFmx2QTc91hK
35 lGOA8SjEtH7iatb68jJLmhw3vBeGCV3EEanH
40 Kijc9/z8U27GnY6BSRbTkXjyMQsqt/AVKo0S
45
50 <s>ANzxWWHwWH5MCaa
55 </ESRP>
60 </SecureLoginChallenge>
65 </AuthResponse>

3. This is a "continuation" of the request from example 1, above, (i.e., the second message sent from the client 22 to the protocol proxy 34).

25 <AuthRequest xmlns="http://ns.s2ml.org/s2ml">
 <ID>urn:PasswordAuthenticatorApplet:e58bd988ee:2</ID>
 <Date>2001-03-29T11:23:651-08:00</Date>
 <InResponseTo>urn:AuthenticationServlet:e58bd98307:1</InResponseTo>
 <Credentials>
 <SecureLogin xmlns="http://sigaba.com/2000/12/sigabanet/fast">
 <ESRP>
 <r>OUqDVi6xfDquAJJoiX6TMQ==</r>
 <proof>Kq4wU50gR8lmM1LWIgsRX6AjdWI=</proof>
 </ESRP>

30

<encrypted>

Jj2hQq0rgYLmuX2Nvml3pdz0m4pADLh7C1AtKDcbjkLWjL3XRbPTRMjEtqe
5/xDzFTpvUr/QbMs1PAs46awwvqHJrIJJj33DDObHa0oFdsJkcuk8oBwoHgswu9UKNhD2+TY
AY5A1XVHYRGgDPw2fNw==

5 </encrypted>
 </SecureLogin>
 </Credentials>
 <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
 <SignedInfo>
 <CanonicalizationMethod algorithm="http://www.w3.org/TR/2000/CR-xml-
10 c14n-20001026"></CanonicalizationMethod>
 <SignatureMethod algorithm="http://www.w3.org/2000/09/xmldsig#hmac-
 sha1"></SignatureMethod>
 <Reference uri="#xpointer(..../../.)">
 <Transforms>
 <Transform algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-
20 20001026"></Transform>
 </Transforms>
 <DigestMethod algorithm="http://www.w3.org/2000/09/xmldsig#sha1">
 <DigestValue>6b2BeIg9NdacKv3icawD6Gc5sQI=</DigestValue>
 </DigestMethod>
 </Reference>
 </SignedInfo>
 <SignatureValue>lnZ21+ccWFSKb+e8vp8FRmF+HfE=</SignatureValue>
25 </Signature>
 </AuthRequest>

4. This is the second response from the protocol proxy 34 to the client 22. This response contains encrypted name assertion and entitlements on the <encrypted> element.

30 <AuthResponse xmlns="http://ns.s2ml.org/s2ml">
 <ID>urn:AuthenticationServlet:e58bd98307:2</ID>

<Date>2001-03-29T11:23:08Z-08:00</Date>
<InResponseTo>urn:PasswordAuthenticatorApplet:e58bd988ee:2</InResponseTo>
<Result>Success</Result>
<ESRP xmlns="http://sigaba.com/2000/12/sigabanet/fast">
5 <proof>fpOoMrfTFPtixO+Py/VnL2wYbK4=</proof>
</ESRP>
<encrypted xmlns="http://sigaba.com/2000/12/sigabanet/fast">
+hTqqvIfr5fjWDnmezmdSPC8ZOMFnTbheg7hRjgL5X8pTzQ5kF/orOxnzx3x9S/J

10

... [Encrypted data omitted for brevity here.] ...

15
16
17
18
19
20

p9BPeVCysg6Yq8iWk3Y6XWHH5/lrzavFn64S5EzLpgGriKySupof4EvOfrdPaP33
</encrypted>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
25 <SignedInfo>
 <CanonicalizationMethod algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"></CanonicalizationMethod>
 <SignatureMethod algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"></SignatureMethod>
 <Reference uri="#xpointer(..../..)"/>
 <Transforms>
 <Transform algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"></Transform>
 </Transforms>
 <DigestMethod algorithm="http://www.w3.org/2000/09/xmldsig#sha1">
 <DigestValue>Rke9WGLhQ1Bwf8yAqHV2p5J2Kz8=</DigestValue>
 </DigestMethod>
 </Reference>
 </SignedInfo>
 <SignatureValue>wvJCJcGhQ/W3UKhoJAGnS2T8nK8=</SignatureValue>
30 </Signature>

</AuthResponse>

5. This is simply a clear-text version of the <encrypted> element in example 4, above.

<NameAssertion xmlns="http://ns.s2ml.org/s2ml">

5 <ID>urn:mechanism/SigAuthServlet:e587bfe6c7@SIGABA.COM:3d</ID>

 <Issuer>mechanism/SigAuthServlet:e587bfe6c7@SIGABA.COM</Issuer>

 <Date>2001-03-29T11:19:980-08:00</Date>

 <Audiences>urn:*</Audiences>

 <ValidityInterval>

10 <NotBefore>2001-03-29T10:19:980-08:00</NotBefore>

 <NotAfter>2001-03-29T19:19:980-08:00</NotAfter>

 </ValidityInterval>

 <AuthData>

 <AuthType>Login</AuthType>

 <UserHandle>logan@sigaba.com@SIGABA.COM</UserHandle>

 <Aliases>logan@gedanken.org@SIGABA.COM,logan@sigaba.com@SIGABA.CO

M,enterprise@sigaba.com@SIGABA.COM,engineering@sigaba.com@SIGABA.COM,enterpri

se-

20 tech@sigaba.com@SIGABA.COM,all@sigaba.com@SIGABA.COM,fastlist@sigaba.com@SI

GABA.COM</Aliases>

 <IdentityToken>

 <dh-public-value xmlns="http://sigaba.com/2001/1/common/security">

 <BigInteger>AOmstRCliTNQ8sbRHlrhVxcBK6SAYWdlMMIyR6MJKrSiU

 /qhZcfYTWSGb0ni9MBDsnkdVBsa+/DA6PpjI45YhsCLi/ZExOMbqKSwLx2wp84Hu+s5Sa9X

25 A+yTcA/WgzRTF+hjErU1fSImgYWd7326zA01D/WMhubYh+XM7nYgKv</BigInteger>

 <BigInteger>Ag==</BigInteger>

 <int>128</int>

 <BigInteger>AMGVIvdII0GwAKE05VLkECE9CrgYLXtNfPxPzNgBLQIp

WCqzXvnP29itL6zkoczpS1Oi+zIL9RFxi37MCkvuNxqCtIykq4XGLAf/PeIPEWNPz9xK3Qvpk

30 Br1yQChIvTuTktLTM+/sQePUwk0LMt/Sy43QDXhyP4Awytucc29k21l</BigInteger>

 </dh-public-value>

5

</IdentityToken>
</AuthData>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
 <SignedInfo>
 <CanonicalizationMethod algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"></CanonicalizationMethod>
 <SignatureMethod algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"></SignatureMethod>
 <Reference uri="#xpointer(..../..)"/>
 10 <Transforms>
 <Transform algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"></Transform>
 </Transforms>
 <DigestMethod algorithm="http://www.w3.org/2000/09/xmldsig#sha1">
 <DigestValue>BnsirjWjSnsNx+ui75eSnCfLK/4=</DigestValue>
 </DigestMethod>
 </Reference>
 15 </SignedInfo>
 <SignatureValue>hjRVhP2sQAwnOZcj/w7WnDxZJuktp2IYX3wF3j8sAEoUkLxuxAkO
 20 SA=</SignatureValue>
 <KeyInfo>
 <KeyValue>
 <DSAKeyValue>
 <P>AOmstRCli!TNQ8sbRHlrhVxcBK6SAYWdlMMIyR6MJKrSiU/qhZ
 25 cfYTWSGb0ni9MBDsNkdVBsa+/DA6PpjI45YhsCLi/ZExOMbqKSwLx2wp84Hu+s5Sa9XA+y
 TcA/WgzRTF+hjErU1fSImgYWD7326zA01D/WMhubYh+XM7nYgKv</P>
 <Q>ANnezYYG5JOCcSNre+pJztsNA2n9</Q>
 <G>AKWsWEe8wT1KkMIIi+u05wZhODK4U0ZnutBGSY+4LtLl++EW
 6E5AcEy8dbR9V4HWU32tQUyZwIikvChkfRXnjRP1/kPsNiAXUCU1AqxL6f1YZIW5zZtXZC
 30 Bs8iSLqF3EOzoUXm5Kqx5TfoTmCvZiM2nYxL6Q9hRZKwgJpC3AnzFbu</G>
 <Y>BPT2HnTOBMfvFZb8UDn6fQ19gFkOsjGGZGGBRRh8Kyr06esp

L34MEavYnwwGW6D1VWbvIuaDHtUfRv6znkwzg0iCtsSnQ2FLD+rpsEQHok7pz+UWDV4L

5u4mhqzWWX3EiJmniOoxQduxqHjsbXm3XMqByWmeJOCVRIVeNa7Msw=</Y>

</DSAKeyValue>

</KeyValue>

5 </KeyInfo>

</Signature>

</NameAssertion>

10 <Entitlement xmlns="http://ns.s2ml.org/s2ml">

<ID>urn:mechanism/SigAuthServlet:e587bfe6c7@SIGABA.COM:3d:0</ID>

<Issuer>mechanism/SigAuthServlet:e587bfe6c7@SIGABA.COM</Issuer>

<Date>2001-03-29T11:19:980-08:00</Date>

<Audiences>urn:*</Audiences>

<ValidityInterval>

<NotBefore>2001-03-29T10:19:980-08:00</NotBefore>

<NotAfter>2001-03-29T19:19:980-08:00</NotAfter>

</ValidityInterval>

<DependsOn>urn:mechanism/SigAuthServlet:e587bfe6c7@SIGABA.COM:3d</Depend

20 sOn>

<AzData>

<ks-info xmlns="http://sigaba.com/2000/12/sigabanet/sigauth">

<ks-info host="minnie.ironsite.com" id="0" ip="63.202.162.58"></ks-info>

</ks-info>

</AzData>

25 <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">

<SignedInfo>

<CanonicalizationMethod algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"></CanonicalizationMethod>

<SignatureMethod algorithm="http://www.w3.org/2000/09/xmldsig#dsa-

30 sha1"></SignatureMethod>

<Reference uri="#xpointer(..../..)"/>

<Transforms>

<Transform algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"></Transform>

</Transforms>

5 <DigestMethod algorithm="http://www.w3.org/2000/09/xmldsig#sha1">

<DigestValue>1yMLZu/99bWEnSIfV1soodcddTk=</DigestValue>

</DigestMethod>

</Reference>

</SignedInfo>

10 <SignatureValue>I6094fFSSCdbRS3HSEEjf2nlyDeapzM/B/NHjJs5y9HX9XvhwfVg
 OoQ=</SignatureValue>

<KeyInfo>

<KeyValue>

<DSAKeyValue>

 <P>AOmstRCliITNQ8sbRHlrhVxcBK6SAYWdlMMIyR6MJKrSiU/qhZ
 cfYTWSGb0ni9MBDsnkdVBsa+/DA6PpjI45YhsCLi/ZExOMbqKSwLx2wp84Hu+s5Sa9XA+y
 TcA/WgzRTF+hjErU1fSImgYWd7326zA01D/WMhubYh+XM7nYgKv</P>

<Q>ANnezYYG5JOCcSNre+pJztsNA2n9</Q>

 <G>AKWsWEe8wT1KkMII+u05wZhODK4U0ZnutBGSY+4LtLl++EW
 6E5AcEy8dbR9V4HWU32tQUyZwIikvChkfRXnjRP1/kPsNiAXUCU1AqxL6f1YZIW5zxtXZC
 Bs8iSLqF3EOzoUXm5Kqx5TfoTmCvZiM2nYxL6Q9hRZKwgJpC3AnzFbu</G> <Y>BPT2HnTOBMfvFZb8UDn6fQ19gFkOsjGGZGGBRRhw8Kyr06esp
 L34MEavYnwwGW6D1VWbvIuaDHtUfRv6znkwzg0iCtsSnQ2FLD+rpsEQHok7pz+UWDV4L
 5u4mhqzWWX3EiJmniOoxQduxqHjsbXm3XMqByWmeJOCVRIVeNa7Msw=</Y>

25 </DSAKeyValue>

</KeyValue>

</KeyInfo>

</Signature>

</Entitlement>